Proceedings of the 2003 IEEE Aerospace Conference, Big Sky MT, March 2003

eSTORM: Enhanced Self Tuning On-board Real-time Engine Model [1]

Tom Brotherton
The Intelligent Automation Corp
Poway, CA 92064
(858) 679-4140
tom.brotherton@iac-online.com

Al Volponi
Pratt & Whitney
East Hartford, CT
(860) 557-2421
volponi@pweh.com

Rob Luppold
Luppold & Associates
West Newton, PA. 15089
(724) 872-5480
rob@luppoldassoc.com

Donald L. Simon
U.S. Army Research Lab
Cleveland, OH
(216) 433-3740
Donald.L.Simon@grc.nasa.gov

Abstract— A key to producing reliable engine diagnostics and prognostics resides in the fusion of different processing techniques. Fusion of techniques has been shown to improve diagnostic performance while simultaneously reducing false alarms. Presented here is an approach that fuses a physical model called STORM (Self Tuning On-board, Real-time engine Model) developed by Pratt & Whitney, with an empirical neural net model to provide a unique hybrid model called *enhanced* STORM (eSTORM) for engine diagnostics. STORM is a piecewise linear approximation of the engine cycle deck. Though STORM provides significant improvement over existing real-time engine model methods, there are several effects that impact engine performance that STORM does not capture. Integrating an empirical model with STORM accommodates the modeling errors. This paper describes the development of eSTORM for a Pratt & Whitney high bypass turbofan engine. Results of using STORM and eSTORM on simulated engine data are presented and compared. eSTORM is shown to work extremely well in reducing STORM modeling errors and biases for the conditions considered.

TABLE OF CONTENTS

1. INTRODUCTION

A key to producing reliable engine diagnostics and prognostics resides in the fusion of sensor data, information, and / or processing algorithms. There are many different approaches that support the development of such systems. These approaches can be generalized into three basic classes. First are physical models. Second are rule-of-thumb systems developed and refined by human engineering and maintenance experts. Third are empirical models that 'learn' from examination of real data that contain nominal and known fault conditions. Each of those techniques has unique strengths and weaknesses. Presented here is an approach that fuses a physical model with an empirical model to provide enhanced diagnostic and prognostic capabilities. Figure 1 shows a high level flow diagram of the hybrid system architecture. Fusion of techniques has been shown to improve detection and classification performance while simultaneously reducing false alarms [1,2]. Fusion of techniques is a way to move towards the utopian goal of perfect detection / classification and zero false alarms.

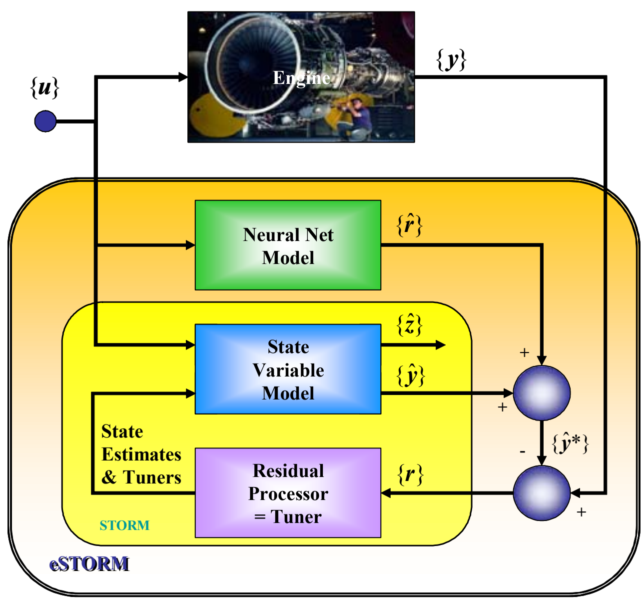

Figure 1 The hybrid model architecture

Here we consider a real-time physics-based model of a large high bypass turbofan engine. The model (called STORM: Self Tuning On-board, Real-time engine Model) is a piecewise linear approximation of the engine cycle deck. STORM contains two primary components: 1) a nominal state variable model (SVM) representation of the engine cycle, and 2) a subsystem that processes the SVM and real engine output differences (*residuals*) to adapt (or "tune") the SVM to off-nominal conditions. A Kalman Filter observer is used to estimate a set of *Tuners* that capture engine

[1] Paper #1023 - This effort is being funded by NASA Dryden
0-xxxx-xxxx-X/01/$10.00/©2003 IEEE

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **01 MAR 2003**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**
4. TITLE AND SUBTITLE **eSTORM: Enhanced Self Tuning On-board Real-time Engine Model 1**		5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)		5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **The Intelligent Automation Corp Poway, CA 92064**		8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)		10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)
12. DISTRIBUTION/AVAILABILITY STATEMENT **Approved for public release, distribution unlimited**		
13. SUPPLEMENTARY NOTES **The original document contains color images.**		
14. ABSTRACT		
15. SUBJECT TERMS		

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**SAR**	**12**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

deterioration as a measure of the percentage of off-nominal operation (or percentage degradation) for major rotating components within the engine gas path. This real-time model can be run at any point in the operational flight envelope.

The STORM concept provides a significant improvement over existing real-time engine model methods and has been adapted to a variety of Pratt & Whitney engines. However, there are several effects that impact engine performance that STORM does not capture. In fact variances often exist between real engine outputs and their corresponding engine Deck counterparts. Currently, STORM's "Tuners" will adapt to account for these modeling errors, but the diagnostic information will be corrupted by the modeling error.

Integrating an empirical model with STORM offers an approach for addressing this class of modeling errors. The empirical model is developed from residuals derived from the true engine outputs and STORM predicted outputs. The empirical model used here is neural network based. The empirical neural network based model is fused with the physics-based STORM model to form a unique hybrid model of the engine. We refer to this model as the *enhanced* STORM model or eSTORM. The objective of including the empirical element is to capture the unmodeled errors as well as specific engine differences so as to eliminate their corruptive impact on the diagnostic information contained in the STORM *Tuners*.

This paper describes the development of eSTORM. Results of using STORM and eSTORM on simulated engine data are presented and compared. eSTORM is shown to work extremely well in reducing STORM modeling errors and biases for the conditions considered.

2. Model Development

Described here is the overall model development starting with STORM, followed by details of the empirical model, and finishing with the integration and "training" to develop the full eSTORM model.

"Real engine" data is required for training and demonstration of the eSTORM concept. There are a variety of problems associated with dealing with real data, the primary one being "is the data any good?" To have control over the data and to address training issues simulated data should be used. However in order to demonstrate the concepts, the model used for simulation of "real engine" data needs to be independent from the one used to develop STORM. For "real engine" simulations we used the D01 Customer Deck to represent an F-117 engine. STORM was developed using the D03 F-117 Simulation Deck. Both the D01 and D03 Decks are physics based models of the real F117 engine. However they are models for different versions of the engine. The difference is similar to that expected to exist between a physical engine and model (due to modeling errors) and allow for the benefit of eSTORM to be demonstrated.

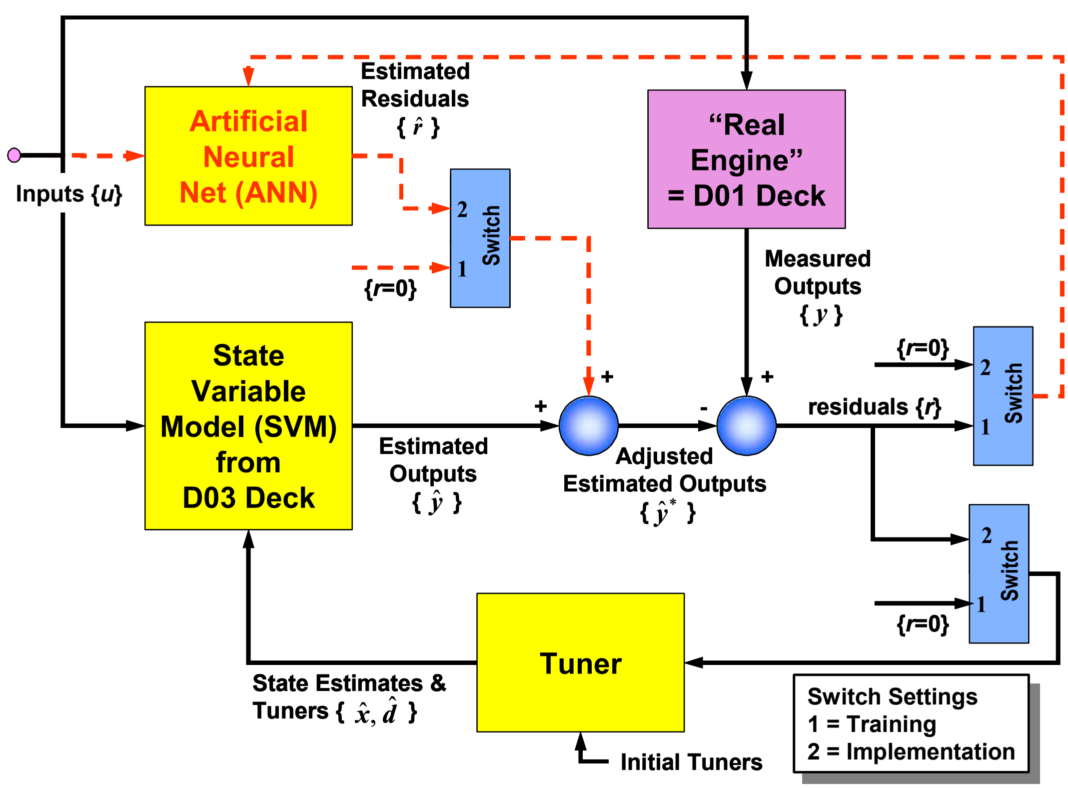

Figure 2 eSTORM system development

To train the system we used the flow diagram shown in Figure 2 with the Switch=1 setting. There are several steps involved in eSTORM training. They are:

1. Run the "Real engine" simulation (i.e. the D01 Customer Deck) to generate *measured outputs* $\{\boldsymbol{y}\}$.

2. Run the nominal STORM SVM with Tuning disabled (i.e., the input residuals are set to $\{\boldsymbol{r}\}=0$) to generate *estimated outputs* $\{\hat{\boldsymbol{y}}\}$. Note: the residual vector is the difference between the measured outputs $\{\boldsymbol{y}\}$ and the estimated outputs $\{\hat{\boldsymbol{y}}\}$.

3. Compute and store the residuals $\{\boldsymbol{r}\}$ along with the inputs $\{\boldsymbol{u}\}$.

4. Train the neural network using the stored residuals and inputs to create an empirical model to predict the *estimated residuals* $\{\hat{\boldsymbol{r}}\}$.

Testing then follows using the trained neural network in the eSTORM system shown in Figure 2 with the Switch=2 setting. This is the same as the system shown in Figure 1. Details of the various steps are given in subsequent sections.

STORM

Figure 3 shows the physics-based STORM model. STORM contains two primary components: 1) a nominal state variable model (SVM) of the Pratt-Whitney engine, and 2) a subsystem that adapts (tunes) the SVM under off-nominal conditions. The SVM uses correction factor theory [3] to cover the full flight envelope. As a baseline, the SVM reflects nominal engine performance at a sea-level-static (SLS) flight condition, but the SVM also accepts engine component deterioration estimates (“Tuners”) and state estimates from the **Tuner** module as inputs. This feature enables the SVM to modify its output estimates to reflect off-nominal engine performance.

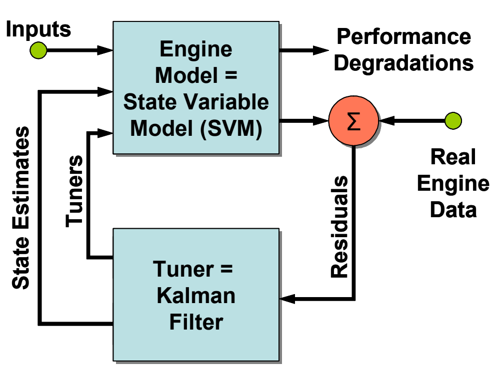

Figure 3 The STORM Physics-Based Model

The **Engine Model** subsystem computes the engine's output response (rotor speeds, pressures, temperatures, airflows, thrust, etc.) to input stimuli (fuel flow, bleeds, etc.). While the baseline model is intended to represent a nominal engine, it also accepts and responds to inputs describing incremental changes to nominal engine component performance (e.g., deterioration). The **Tuner** module generates estimates of these incremental performance shifts and the **Engine Model** incorporates these effects into the engine’s predicted output response vector in Figure 3.

The SVM used in this application is derived from a detailed, component-level aerothermodynamic model of the engine cycle. The resulting onboard model consists of two pieces:

1. A piecewise linear dynamic state variable model, that operates in corrected parameter space. The dynamic model contains seven states (two rotor speeds and five metal temperatures), thirteen inputs (engine controls, installation effects, and deterioration effects), and twenty outputs (pressures, temperatures, airflows, and thrust). The reference flight condition for the dynamic model is sea-level-static (SLS).

2. A tabulated set of steady-state points (commonly referred to as the SVM base-points) that capture the inherent large-signal nonlinear behavior of the engine cycle at a reference flight condition. The reference flight condition is SLS. Each element of the state, input, and output vectors contain an associated base-point.

There are several second order effects that are not captured by the SVM that have an impact on gas turbine rotating machinery performance. These effects include:

- Tip clearance effects on flow capacity and efficiency,
- Reynolds effects,
- Blade untwist effects on fan performance.

Since these three phenomena are not modeled by the SVM, their impact on model outputs will be reflected in the Tuner module’s off-nominal performance estimates. Though this adaptation mechanism will reduce the SVM output estimation error, the diagnostic information reflected in the Tuner outputs will become corrupt. One of the objectives of the empirical model addition is to capture these unmodeled performance effects and eliminate their impact on the diagnostic information contained in the onboard model Tuners.

A standard set of so-called correction factors are used to adjust the SVM to points in the engine’s operational flight envelope that differ from the SLS reference point [3]. Though correction factor theory is predominately based on steady-state phenomena, the concept has been extended to cover the time-scale variations experienced at various points in the operational flight envelope during dynamic engine operation. In the work presented here, the effect of correction factor accuracy was not an issue since all the performance evaluations of STORM with empirical model addition were performed at the SLS flight condition. Although not an issue here, future work will address additional flight conditions where correction factor accuracy will be important.

An additional feature incorporated into the engine model is the *Deterioration Performance* subsystem. This subsystem processes the Tuner input vector in order to separate the engine deterioration performance estimates from the control input offsets. In this application, the Tuner vector contains six elements, five incremental component performance estimates (a vector that is output from the SVM that measures a component's performance deterioration), and a fuel flow bias offset estimate. The fuel flow compensation is mandatory because of the general inaccuracy of fuel flow measurement systems. Since fuel flow is the dominant input to the engine model and inputs are assumed to be deterministic, a fuel flow offset correction must be applied to avoid significant corruption of the engine performance tuners.

Online STORM Update

The recursive estimation technique used in this application is based on the linearized Kalman filter concept [4]. The model linearization approach partitions a state variable model (SVM) into steady state (base-point) and dynamic components. The steady state model schedules were based on low rotor speed and ram pressure ratio. The piecewise-linear dynamic models were scheduled as a function of low rotor speed at a predefined reference flight condition. In the F-117 application, the reference flight condition corresponds to Sea-Level-Static (SLS). Using the SVM framework, a Kalman filter design model is formulated and linked to the functional specification of the real-time algorithm that updates the SVM as the engine lifecycle evolves. The computational tools for constructing a filter design model and synthesizing a Kalman filter utilized the capabilities of the MATLAB/SIMULINK™ environment.

Since the advent of full-authority digital engine controls (FADEC) in the early 1980's, the engine control system design community has pursued the concept of a *robust* engine model for a myriad of reasons. Enhancement of performance in engine control systems, fault detection, isolation, & accommodation systems, and diagnostic systems are obvious focal points of this interest. However, the largest obstacle to incorporating model-based designs in gas turbine systems has been the accuracy issue, i.e., how precisely does the model track actual engine behavior. Two factors make significant contributions to this problem. First of all, we know that there is a degree of uncertainty that exists between any model and the actual engine. Parametric uncertainties, ignored dynamics, measurement uncertainties, etc. are realities that designers must cope with in creating model-based systems. Design techniques like robust H_∞ estimation can be used to systematically include these effects into the synthesis process. Unfortunately, methods used to achieve robustness to these uncertainties extract penalties on estimation performance, especially where insensitivity to large uncertainties must be accommodated. Another factor influencing the robust model design in gas turbine application is the desire to track the component life cycle evolution of these systems. Integrating this information into the model will significantly enhance the predictive fidelity of the model, but it also provides a database for diagnosing and maintaining the gas turbine system. Therefore, an alternative approach to solving the robust engine model problem is to estimate a set of signals that reflect the differences between the nominal model and the actual engine. This set of signals is then used to update the nominal engine model so that estimation performance is maintained. The Pratt & Whitney STORM implements this approach.

The implementation of the STORM algorithm requires a recursive estimation technique. Historically, the technique chosen in gas turbine applications has been the linearized Kalman filter [5]. There are several reasons for this. First of all, other candidate nonlinear estimators have complexity issues and require computational resources that are beyond the capabilities of current microprocessors used in real-time gas turbine applications. Secondly, techniques like the extended Kalman filter require explicit (closed-form) knowledge of the nonlinearities in the state and output equations. In current gas turbine modeling technology, this information is not available. Hence, a considerable amount of work has been done in the last twenty-five years to develop accurate linearized SVMs for model-based control and diagnostic applications. The basic concept is straightforward. The designer needs to build a steady-state model that accurately predicts nominal performance. This model captures the nonlinearities of the component maps and other major nonlinearities associated with the aerothermodynamic processes. Based on this steady-state operating line, a perturbational dynamic model can be built that reflects the off-nominal behavior of the system.

This approach has been successfully applied in a variety of military engine programs. These programs demonstrated the concept for low-bypass, augmented turbofan cycles. For the past eight years, Pratt has been transferring STORM technology to high-bypass, turbofan engine applications. The design concepts described here reflect the lessons learned in porting the STORM technology to high-bypass turbofan engines [6].

The steady-state model contains a set of base-points, $\{\mathbf{x_b},\mathbf{u_b},\mathbf{y_b},\mathbf{z_b}\}$, tabulated for a series of points along the engine operating-line from idle to full power at a given reference flight condition. Throughout the remainder of this section, $\mathbf{x}$ refers to the engine state vector, $\mathbf{u}$ to the *deterministic* engine input vector, $\mathbf{y}$ to the measured engine outputs, and $\mathbf{z}$ to the uninstrumented engine outputs. The reference flight condition is arbitrary, but all STORM applications to date have selected the SLS flight condition as the reference point. The table lookup procedure for the base-point model requires a bivariate linear interpolation algorithm. The independent variables in this procedure are corrected low rotor speed and ram pressure ratio (Pt2/Pamb). Corrected low rotor speed, NLc, is a state in the SVM, however, the ram-ratio is based on filtered measurements from the gas turbine instrumentation suite and is directly correlated with Mach number. The steady-state model used in this application contains thirty-five NLc points at five different ram pressure ratios.

The dynamic engine model reflects perturbational effects about a steady-state equilibrium condition. The complete piecewise linear model is a collection of points generated along the nominal operating line from idle to full power. In this application, the set of linearized models is scheduled on the corrected low rotor speed (NLc) state variable. Hence, the perturbational points that characterize the linearized model coincide with the steady-state model at the Sea-Level Static flight condition. The set of engine partials generated for this application differs somewhat from those used in the conventional control system design process. For the Kalman filter design model, we need to include the impact that component deterioration has upon the engine states and outputs. One approach to modeling this effect treats component deterioration as a system input; however, unlike the externally supplied engine inputs, we cannot directly measure component deterioration, and therefore, these inputs are unknown. We can express in mathematical terms the state and output equations associated with this unknown input problem as

$$\begin{aligned}\delta\dot{\mathbf{x}}(t) &= \mathbf{A}\delta\mathbf{x}(t)+\mathbf{B}_1\delta\mathbf{u}(t)+\mathbf{B}_2\mathbf{d}(t)\\ \delta\mathbf{y}(t) &= \mathbf{C}_1\delta\mathbf{x}(t)+\mathbf{D}_{11}\delta\mathbf{u}(t)+\mathbf{D}_{12}\mathbf{d}(t)\\ \delta\mathbf{z}(t) &= \mathbf{C}_2\delta\mathbf{x}(t)+\mathbf{D}_{21}\delta\mathbf{u}(t)+\mathbf{D}_{22}\mathbf{d}(t)\end{aligned} \tag{1}$$

In these equations the δ-symbol explicitly indicates that the quantity it precedes is a small signal perturbation from an equilibrium condition, e.g., $\delta\mathbf{x}=\mathbf{x}-\mathbf{x}_b$. By definition the $\mathbf{d}$-term, the deterioration input, represents a small deviation from nominal component performance and therefore does not require the δ-symbol designation. The upper-case symbols represent the Jacobians of the nonlinear engine model.

Given an engine model like the one in Eq (1), we need to formulate a methodology for dealing with the unknown input $\mathbf{d}(t)$. There are two approaches to solving this problem. In one case, we can formulate a state estimation problem that decouples the unknown inputs from the state estimation error, i.e., the so-called Unknown Input Observer (UIO) problem [7]. This approach attempts to preserve estimation accuracy without directly calculating the unknown input vector. However, knowing $\mathbf{d}(t)$ has intrinsic value in an engine diagnostic sense. Hence, the second approach of directly estimating the level of off-nominal engine performance is preferable in STORM applications.

To solve this estimation problem, we need to assume a model that fits the deterioration process and can also be used effectively within the Kalman filter design framework. If we temporarily exclude foreign object damage (FOD) from the deterioration model, then an accurate depiction of the engine life-cycle aging is the so-called *slowly varying constant* model [8]. To properly formulate this model requires a stochastic (random) process framework. In this framework, integrating a white noise source produces a slowly varying, random constant, i.e., the so-called random walk or Wiener process. Expressing this model in terms of a state space realization results in

$$\begin{aligned}\dot{\mathbf{x}}_d(t) &= \mathbf{w}(t)\\ \mathbf{d}(t) &= \mathbf{x}_d(t)\\ E\{\mathbf{w}(t_1)\mathbf{w}^T(t_2)\} &= \mathbf{Q}\delta(t_1-t_2)\end{aligned} \tag{2}$$

where $\mathbf{w}(t)$ denotes a white noise vector process with an intensity matrix of $\mathbf{Q}$, the symbol E{•} represents the statistical expectation operator, and δ(•) is a delta function. Since the function $\delta(t_1-t_2)$ is 0 unless $t_1 = t_2$, the white noise process is uncorrelated in time. Moreover, a white noise process is a Gaussian, zero-mean process by definition. Therefore, specifying the $\mathbf{Q}$-matrix completely characterizes the random process for our model of component deterioration. For this study, we modeled the deterioration of the major rotating machinery components in the gas path, i.e., fan, low-pressure compressor (LPC), high-pressure compressor (HPC), and high-pressure turbine (HPT).

In both commercial and military gas turbine applications, using the Bill-of-Material (control) instrumentation suite produces a measurement vector that includes rotor spool speeds, temperatures, and pressures at key engine stations. Each of these measurements contains a non-deterministic component, i.e., noise. Hence, the engine output measurement relationship in Equation (1) needs to be modified to include this term, so that

$$\begin{aligned} \delta \mathbf{y}(t) &= \mathbf{C}_1 \delta \mathbf{x}(t) + \mathbf{D}_{11} \delta \mathbf{u}(t) + \mathbf{D}_{12} \mathbf{d}(t) + \mathbf{v}(t) \\ E\left\{\mathbf{v}(t_1)\mathbf{v}^T(t_2)\right\} &= \mathbf{R}\delta(t_1 - t_2) \\ E\left\{\mathbf{w}(t)\mathbf{v}^T(t)\right\} &= 0 \quad \forall t \end{aligned} \tag{3}$$

The measurement noise is denoted by $\mathbf{v}(t)$ and is assumed to be a white, Gaussian, vector process with a covariance matrix of $\mathbf{R}\,\delta(t)$. Note that the random process driving the deterioration input, $\mathbf{w}(t)$, and the measurement noise are assumed to be uncorrelated in this problem formulation.

To formulate the Kalman filter design model we merged the engine and deterioration models to form one model. The resultant augmented state vector contains the normal engine states as well as the deterioration inputs, $\mathbf{d}(t)$. A compact way of representing this mathematically in the state and output equations is

$$\begin{aligned} \begin{bmatrix} \delta \dot{\mathbf{x}}(t) \\ \dot{\mathbf{d}}(t) \end{bmatrix} &= \mathbf{A}_{KF} \begin{bmatrix} \delta \mathbf{x}(t) \\ \mathbf{d}(t) \end{bmatrix} + \mathbf{B}_{KF}\, \delta \mathbf{u}(t) + \mathbf{w}(t) \\ \delta \mathbf{y}(t) &= \mathbf{C}_{KF} \begin{bmatrix} \delta \mathbf{x}(t) \\ \mathbf{d}(t) \end{bmatrix} + \mathbf{D}_{11}\, \delta \mathbf{u}(t) + \mathbf{v}(t) \\ \delta \mathbf{z}(t) &= \mathbf{M}_{KF} \begin{bmatrix} \delta \mathbf{x}(t) \\ \mathbf{d}(t) \end{bmatrix} + \mathbf{D}_{21}\, \delta \mathbf{u}(t) \end{aligned} \tag{4}$$

The *KF* subscript that appears in Eq (4) denotes the composite Kalman filter design matrices. Details of this formulation can be found in [6].

Using the augmented state equations in Eq (4), we now have a system model that the Kalman filter design machinery can use for producing a recursive state estimator. The objective is to estimate $\mathbf{x}(t)$ and $\mathbf{d}(t)$, and then, to compute $\mathbf{y}(t)$ and $\mathbf{z}(t)$ using the output equations. The Kalman filter formulation minimizes the mean-square state estimation error induced by random disturbances and measurement noise in *linear* systems. In more generic terms, the Kalman filter provides a systematic framework for establishing a trade-off between measured information and a process model so that an optimal linear estimate of the process outputs can be produced.

In order to achieve this performance goal, the Kalman filter generates an output error, i.e., the so-called filter residual vector, which is defined by

$$\begin{aligned} \mathbf{r}(t) &= \mathbf{y}(t) - \hat{\mathbf{y}}(t) \\ &= \mathbf{y}(t) - \left[\mathbf{C}_1 \delta \hat{\mathbf{x}}(t) + \mathbf{D}_{12} \hat{\mathbf{d}}(t) + \mathbf{D}_{11} \delta \mathbf{u}(t) + \mathbf{y}_b\right] \end{aligned} \tag{5}$$

where the *hat* symbol above $\mathbf{y}$, $\mathbf{x}$, and $\mathbf{d}$ denotes the Kalman filter estimate of the output measurement, engine state, and tuner vectors, respectively. Note that the second equality in Eq (5) expresses the output measurement estimate in terms of the augmented state estimates, the deterministic input vector, and the output base-point, $\mathbf{y}_b$. The Kalman filter applies a weighting factor to the residual vector and then feeds back this error signal to the state equation, i.e.,

$$\begin{aligned} \delta \dot{\hat{\mathbf{x}}}(t) &= \mathbf{A} \delta \hat{\mathbf{x}}(t) + \mathbf{B}_1 \delta \mathbf{u}(t) + \tilde{\mathbf{B}}_2 \underbrace{\hat{\mathbf{d}}_P(t)}_{Performance} + \underbrace{\mathbf{K}_1 \mathbf{r}(t)}_{Deriv.Bias} \\ \underbrace{\hat{\mathbf{d}}(t)}_{Tuners} &= \int_0^t \mathbf{K}_2 \mathbf{r}(\tau) d\tau \\ \underbrace{\hat{\mathbf{d}}(t)}_{Tuners} &= \begin{bmatrix} \underbrace{\hat{\mathbf{d}}_P(t)}_{Performance} \\ \underbrace{\hat{d}_I(t)}_{Input} \end{bmatrix} \\ \delta \hat{\mathbf{x}}(t) &= \hat{\mathbf{x}}(t) - \mathbf{x}_b \\ \delta \mathbf{u}(t) &= \mathbf{u}(t) - \mathbf{u}_b + \underbrace{\hat{\mathbf{d}}_I(t)}_{Input} \end{aligned} \tag{6}$$

$\tilde{\mathbf{B}}_2$ differs from $\mathbf{B}_2$ of equation (1) in that the columns associated with the input biases have been removed. Hence, the state derivative prediction will be based not only on the current state and input, but also on the weighted, output estimation error from the previous state update. The weighting factor, $\mathbf{K} = \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \end{bmatrix}$, is the so-called Kalman filter gain matrix.

Empirical Model Development

The empirical model used here is a neural network. Artificial neural networks (ANN) are an attempt to model the brain by the dense interconnection of a large set of simple processing elements. Neural nets have proven useful in a variety of areas: detection, classification, multidimensional function approximation, and predictive modeling of data. They are ideal for developing non-linear models to map input data to outputs. They can be used for classification-base diagnostics as well as prognostics.

Neural nets are "trained" by presenting examples of input/output pairs of data. For most applications, the output data has been “labeled” as to the correct class or function response. The parameters in the neural net are adjusted during training until the neural net classification performance reaches an acceptable level

The neural net in eSTORM is used to solve a function approximation problem. The neural net forms an empirical model for the (possibly non-linear) transfer function between the inputs $\{\mathbf{u}\}$ and outputs $\{\mathbf{r}\}$ of the system. Following Narendra [9], we develop a non-linear autoregressive, moving-average (NARMA) model to solve the system identification problem. The general NARMA model takes the form:

$$\hat{r}^k(t) = \Psi[\mathbf{r}(t-1),\ldots\mathbf{r}(t-p),\mathbf{u}(t-1),\ldots,\mathbf{u}(t-q)] \qquad (7)$$

where

$$\mathbf{r}(t) = \begin{bmatrix} r^1(t) \\ \vdots \\ r^{\mathbf{nout}}(t) \end{bmatrix} \qquad \mathbf{u}(\mathbf{t}) = \begin{bmatrix} u^1(t) \\ \vdots \\ u^{\mathbf{nin}}(t) \end{bmatrix} \qquad (8)$$

and $\hat{r}^k(t)$ is the estimate of the *k*-th residual output at time t+1, $r^m(t)$ is the *m*-th residual output at time *t*, and $u^n(t)$ is the *n*-th input at time t. With no loss of generality each output is modeled separately. There are a total of nout outputs and nin inputs. *p* is the order of the autoregressive portion of the model and *q* is the order of the moving average part. Ψ is a function that represents the particular type of processing applied to its' arguments to generate the target outputs. We have examined three choices for Ψ: 1) multi-layer perceptron (MLP) neural network, 2) radial basis function (RBF) neural network, and 3) a support vector machine used for regression (SVR) with a Gaussian kernel [10]. The results that gave the minimum mean square error were found using the MLP neural network. Only MLP results are reported here.

For the development we used the NETLAB toolbox that is available on the web [11] for training of the neural networks and initial validation of the processing. For the eSTORM implemented in the SIMULINK graphical language, we have used some of the functions that are in the MATLAb neural net toolbox. This implementation produces code with much faster running speeds when compared to the NETLAB implementation. Note that the functions required for the MLP neural net implementation can be synthesized with standard Simulink blocks.

Testing

Testing was performed using the trained neural network in the eSTORM system shown in Figure 1. In this study, the magnitude of the $\hat{\mathbf{d}}_P(t)$ performance degradation vector provides a metric on the impact made by the empirical model in the eSTORM system. This is demonstrated by executing the simulation with and without the neural net compensation active.

3. SYSTEM TRAINING

The training procedure described above was used to develop an eSTORM system for approximately piecewise-linear throttle operations at sea level static (SLS) conditions. The "real engine" was simulated using the D01 Customer Deck. STORM’s input requirements include:

1. eight external engine control signals,
2. eight engine output measurements,
3. three inlet condition measurements,
4. STORM enable indicator.

The eSTORM system in this initial phase of development was targeted toward piecewise-linear (stationary) throttle operations. In order to evaluate a wide range of 'stationary' throttle operations, data was generated using very slow throttle movements to ramp the throttle from idle to maximum and back to idle. The training data throttle operation started at the minimum throttle value with a 15 second dwell time at the initial throttle setting. After the elapsed dwell time, the throttle was increased from idle to maximum power over a 900 second interval (or 15 minutes). The throttle was then held at maximum power for 100 seconds, followed by a 900 second (15 minute) ramp back down to the idle power level.

Figure 4 shows the inputs used to generate the eSTORM training data. These plots indicated that three of the external engine control inputs are non-zero, one (the 14th stage bleed) is computed to be zero, and four are (set to) zero. The time histories of the three nonzero inputs (fuel flow, stator vane angle, and station 2.5 bleed) are not linearly related.

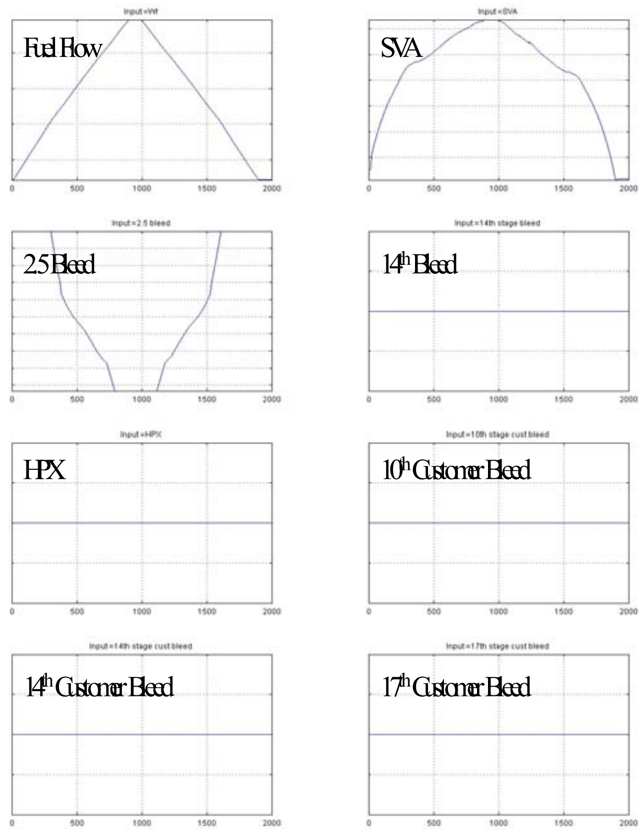

Figure 4 Training inputs

The engine control system inputs shown in Figure 4 were used to drive the STORM SVM with the Tuning subsystem disabled (i.e., the STORM residuals are set to zero).

Figure 5 shows the engine outputs predicted by the D01 Customer Deck overlaid with the STORM SVM estimates. The results in Figure 5 indicate that the SVM outputs with Tuning disabled do not track the D01 deck outputs very well. However this is as to be expected since the SVM is based on the D03 Simulation Deck.

To reduce training data variation, the residuals were smoothed prior to training. Smoothing was accomplished using a low pass filter. Smoothed data results in neural networks that have fewer nodes and better performance (smaller mean square training error) then networks that would be required to model the original data.

Overlaid plots of the raw and smoothed residuals produced in this test case are shown in Figure 6. These plots indicate that the residuals can become quite large. It is interesting to note that the residuals appear to be somewhat symmetric about the full throttle plateau; however upon closer examination there are 'bias' differences in comparing the residuals for the throttle up operation with the throttle down operation. These bias differences can be seen in the input signals as well. The bias indicates that our attempt to approximate a piecewise linear signal with very slow throttle movements was not entirely successful. If our piecewise linear assumption had been correct, then the throttle up and throttle down data should appear symmetric for any particular throttle setting.

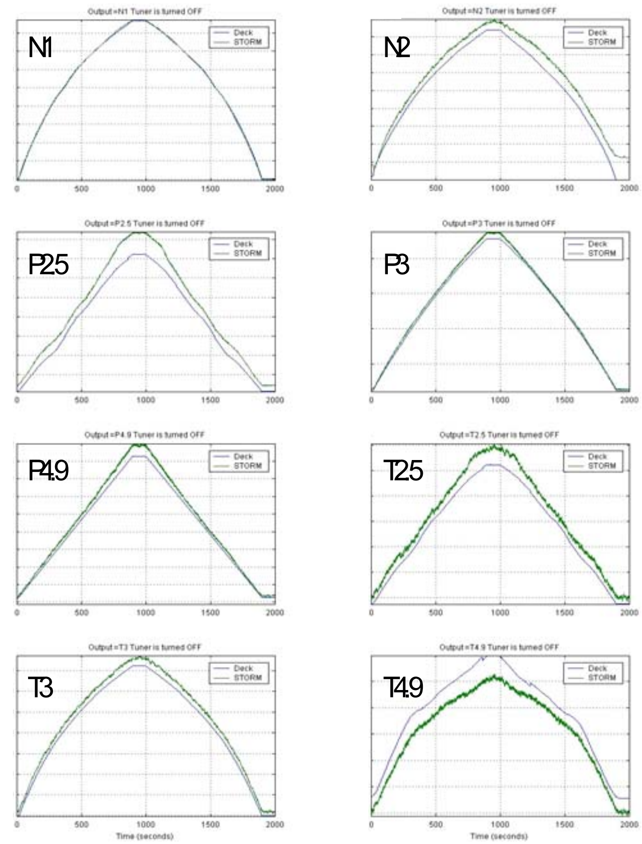

Figure 5 Simulated engine outputs and STORM estimates with tuning disabled

The smoothed residual data shown in Figure 6 was used for training the neural net compensator.

We experimented with several different values of p, q and the number of hidden units in the MLP while developing the empirical model. It was found that small values of p and q produce good results. However, when the ARMA/MLP model with p=1 and q=2 was implemented in eSTORM, the overall results were not very good. The autoregressive component gives rise to a feedback path that uses the same output residuals as STORM. Thus, STORM and the neural net compensator compete against each other in trying to null the residual vector. Determining a systematic methodology for setting gains and time constants in the neural net compensator so that it doesn't interfere with the basic operation of STORM will be investigated in the future.

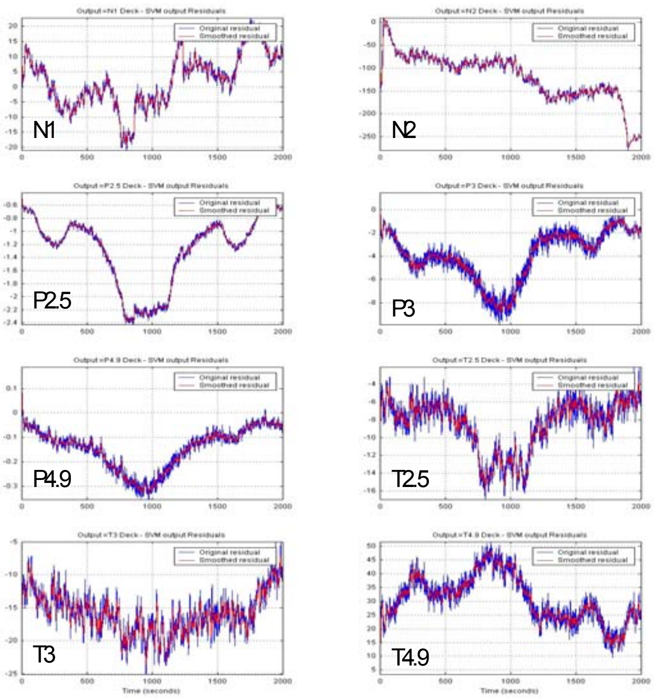

Figure 6 Residuals and smoothed residuals

An alternative design to the ARMA is to remove the autoregressive component of model and only use the smoothing effect of the moving average estimator. One possible candidate design is *p*=0, *q*=3, and the number of hidden units set to 25.

Figure 7 displays the model fit and resulting residuals for one of the outputs associated with this MA/MLP empirical model.

As seen in Figure 7 only the slower moving mean value component of the residual time series is modeled using the MA/MLP. The removal of coarse modeling error component was shown to be sufficient for stabilizing the STORM $\hat{\mathbf{d}}_P(t)$ estimates to acceptable levels.

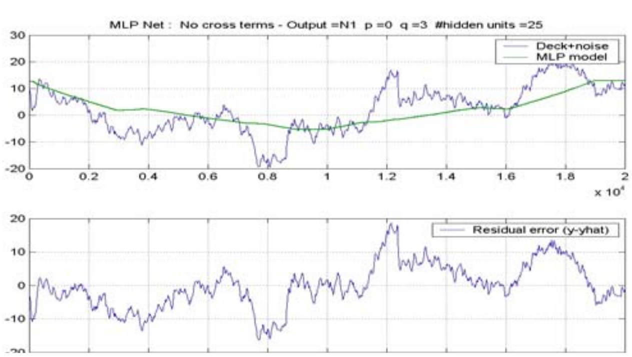

Figure 7 Training results: MLP *p*=0 *q*=3 #basis units=25

The *p*=0 (i.e. there is no AR part), *q*=3, #units=25 MLP neural net regression model was inserted into a candidate eSTORM design. A separate neural net is trained for each of the outputs. The neural net inputs are constructed from just the lagged samples of the engine inputs, i.e.

$$\hat{r}^k(t) = \Psi\left[\mathbf{u}(t-1),\mathbf{u}(t-2),\mathbf{u}(t-3)\right] \qquad (9)$$

Figure 8 shows the comparison plots of eSTORM outputs versus the "real engine" outputs. These results indicate that the eSTORM outputs closely track the simulated engine outputs.

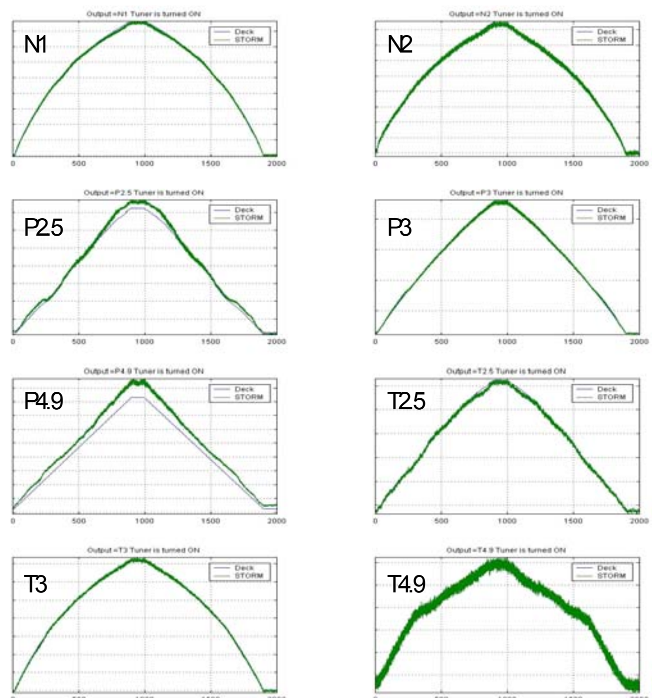

Figure 8 eSTORM Outputs

4. RESULTS

We use the STORM / eSTORM tuner outputs (i.e. the $\hat{\mathbf{d}}_P(t)$ performance degradation vector) to assess the value added of including the empirical model with STORM to form the hybrid eSTORM model. STORM is operated with Tuning enabled. Note that the STORM results under these conditions should NOT be very good. STORM compensates for the differences between the D01 Customer Deck engine simulation and the D03 Simulation Deck by apportioning the apparent off-nominal performance to the $\hat{\mathbf{d}}_P(t)$ estimates. Also note that the STORM design used in this project is not yet mature for this application. A more refined version of the STORM algorithm will be available in future work. The

STORM results presented here are not an indication of how well STORM could work in a typical application. Rather, its purpose is to demonstrate the utility of including the neural net compensation in eSTORM.

A test data set was generated to process through the STORM and eSTORM systems for algorithm validation purposes. The test data set was similar to the training data in that it included two slow throttle movements. However in the test data set, the engine power condition was initially set to the maximum throttle setting, ramped down to idle over a 15-minute interval, remained at idle for 100-seconds and then ramped back to the maximum setting over a 15-minute interval. Different noise realizations were also inserted into the eSTORM inputs. As with the training data, only three of the eight engine control inputs are non-zero

Figure 9 shows the set of $\hat{\mathbf{d}}_P(t)$ "performance degradations" calculated by STORM as it attempts to track the "real engine" outputs. The $\hat{\mathbf{d}}_P(t)$ represent the incremental change in component efficiency required to match model outputs to real engine outputs, in other words the difference between nominal and off-nominal as defined by the D03 SVM. In this case the $\hat{\mathbf{d}}_P(t)$ are forced to become quite large (up to 15% absolute error with about a 5% average absolute error) and must assume unrealistic values in several cases. Again this is primarily due to the D03 SVM and D01 Customer Deck mismatch as well as the immaturity of this particular STORM design. However these results form a good starting point for comparison of $\hat{\mathbf{d}}_P(t)$ values calculated after the application of the neural network compensation.

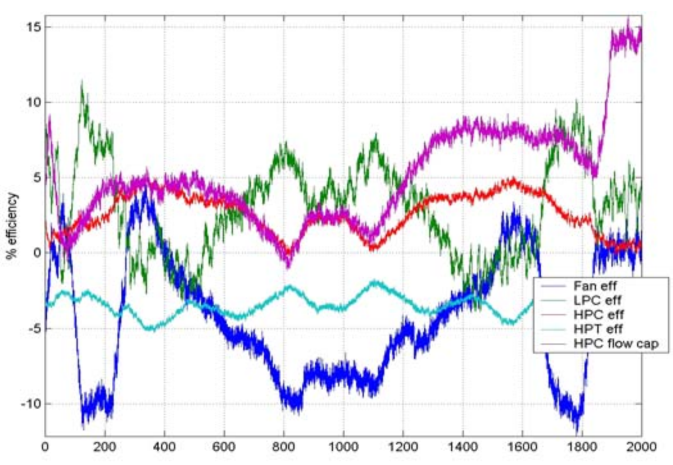

Figure 9 STORM $\hat{\mathbf{d}}_P(t)$ (no neural network compensation)

eSTORM was implemented using the neural networks developed for the up-down throttle movements and applied to the test data set. Figure 10 shows the resulting $\hat{\mathbf{d}}_P(t)$ vector. As can be seen when comparing the $\hat{\mathbf{d}}_P(t)$ of Figure 9 with Figure 10 there is a substantial improvement. The eSTORM $\hat{\mathbf{d}}_P(t)$ are essentially zero-mean, the desired result.

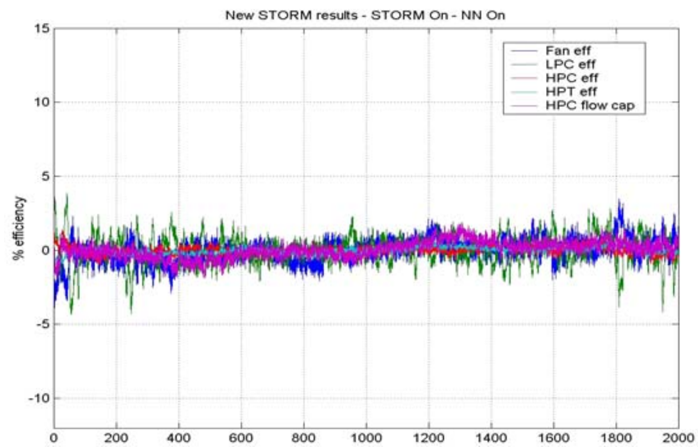

Figure 10 eSTORM $\hat{\mathbf{d}}_P(t)$

Simulated Engine Degradation

To determine the impact of the neural net compensation when engine degradations are present, a second simulation was considered. In this simulation the throttle setting was set to a constant value (i.e. there was no throttle movement through out the simulation) and a 5% degradation was added to both fan and HPC components.

Figure 11 shows the resulting performance degradations found with eSTORM when a simulated 5% degradation is added to the fan. The degradation is turned on at the mid point in the data. As seen the induced degradation is tracked by eSTORM very well.

Figure 12 shows a similar plot, however it is the HPC efficiency that is degraded 5%. As with the previous figure, the degradation is turned on in about the middle of the plot.

Figure 13 shows the results of including both a 5% degradation in the fan performance (turned on at about 600 seconds into the data) followed by the addition of a 5% degradation in the HPC efficiency (turned on at about 1200 seconds into the data).

As seen, eSTORM has the desired performance. The neural net compensation does indeed baseline the STORM processing. However it does not over compensate by removing STORM's ability to track degradations. eSTORM tracks the degradations very well.

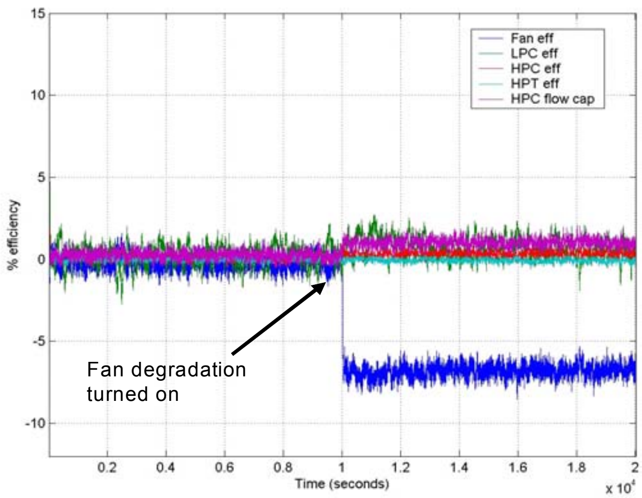

Figure 11 eSTORM $\hat{\mathbf{d}}_P(t)$ with 5% degraded fan performance

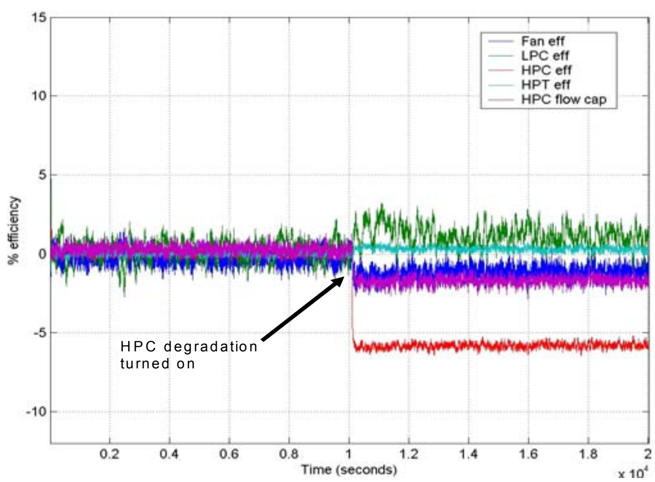

Figure 12 eSTORM $\hat{\mathbf{d}}_P(t)$ with 5% degraded HPC performance

5. SUMMARY AND RECOMMENDATIONS

eSTORM is a hybrid model that fuses / integrates a physics based engine model with an empirical neural net based model. Details of STORM, empirical models, and eSTORM development have been presented. Comparative results of using STORM and eSTORM on simulated engine data show that the hybrid approach works extremely well in reducing STORM modeling errors and biases for the conditions considered. However that compensation does not impact STORM's ability to track degradations in the data.

The results presented here are preliminary. They deal only with steady-state simulated "good" engine in steady state conditions. In future work for NASA we will include transient engine operation as well as real engine data.

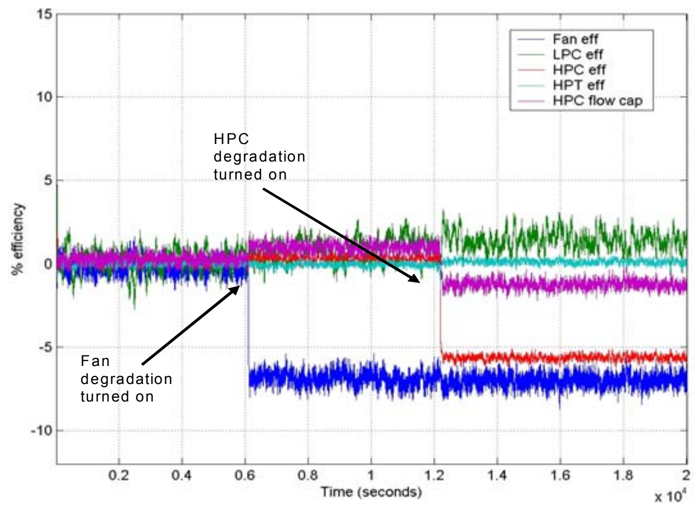

Figure 13 eSTORM $\hat{\mathbf{d}}_P(t)$ with 5% degradation of both the fan and HPC performance

[1] T. Brotherton and E. Mears, "Applications of Neural Nets to Feature Fusion," The 26th Asilomar Conf. on Signals, Systems and Computers, Pacific Grove, CA, Oct. 1992.

[2] T. Brotherton and R. Mackey, "Anomaly Detector Fusion Processing for JSF", *The 2001 IEEE Aerospace Conference, March 2001*, Big Sky, MT.

[3] A.J. Volponi, "Gas turbine parameter corrections" Journal of Engineering for Gas and Power, Vol. 121, Oct. 1999.

[4] P.S. Maybeck, **Stochastic Models, Estimation, & Control: Volume 2**, Academic Press, 1982, pp. 39-59.

[5] R.H. Luppold, J.R. Roman, G.W. Gallops, L.J. Kerr, "Estimating In-Flight Performance Variations Using Kalman Filter Concepts", AIAA-89-2584, AIAA/SAE/ASME/ASEE 25th Joint Propulsion Conference, July 10-12, 1989, Monterey, CA..

[6] R. Meisner, R.H. Luppold, A. Turveskiy, **Advanced Subsonic Technology Program: Final Report**, NASA Contract NAS3-27727.

[7] J. Chen, R.J. Patton, H.Y, Zhang, "Design of Unknown Input Observers and Robust Fault Detection Filters", International Journal of Control, Vol 63, No 1, 10 January 1996, pp. 85-105.

[8] H. Kwakernaak, R. Sivan, **Linear Optimal Control Systems**, Wiley-Interscience, 1972, Ch. 4.

[9] K.S. Narendra, "Neural networks for control: Theory and practice", Proc. of the IEEE, vol. 84, No. 10, Oct 1996.

[10] T. Joachims, Making large-Scale SVM Learning Practical. Advances in Kernel Methods - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999 (http://svmlight.joachims.org)

[11] I.T. Nabney, NETLAB – Algorithms for Pattern Recognition, Springer-Verlag, 2002 (http://www.ncrg.aston.ac.uk/netlab)

Tom Brotherton *received his B.S. degree from Cornell University (1974), an M.A.Sc. from the University of Toronto (1976) and the Ph.D. from the University of Hawaii (1982) all in electrical engineering. He was an assistant professor in the Information and Computer Sciences Department at the University of Hawaii in 1983 and with the Orincon Corp from 1983-1999. He is currently a VP of the Intelligent Automation Corp. (IAC) in San Diego, CA. IAC is involved with the development of aircraft and related equipment monitoring software and hardware. Dr. Brotherton is also on the editorial board for the IEEE Press series on Biomedical Engineering. His interests are in the development of adaptive signal and data fusion techniques for machine condition, fault monitoring, and prognosis.*

Al Volponi *received his B.S. and M.S. degrees from Pratt Institute (1971-2), and Ph.D. from the Adelphi University (1977), all in mathematics. He has been with United Technologies since 1978, initially with the Hamilton Sundstrand Division as a Senior Principal Engineer in the engine controls group and currently holds the position of Fellow in Engine Diagnostics at Pratt & Whitney. His interests are in the area of propulsion health management where he is active in the development of engine performance diagnostic systems. Dr. Volponi was awarded the Manly Memorial Medal by the SAE in 1992 and is currently Chairman of the IGTI Controls, Diagnostics and Instrumentation Committee.*

Rob Luppold *received a B.S. degree in Electrical Engineering from Penn State University (1973) and an M.S. degree in Aeronautics & Astronautics Engineering from MIT (1983). He has held engineering staff positions with Raytheon Missile Systems, C. S. Draper Laboratory, and United Technologies Research Center. Currently, Rob is president and chief executive officer of Luppold & Associates, Inc (L&A), a small business enterprise with an emphasis on the software implementation of control & digital signal processing systems. He has authored numerous papers, reports, and presentations on auto-tuning control systems, self-tuning on-board gas turbine models, analytic fault detection, isolation, & accommodation systems, and the evaluation of fault-tolerant control system reliability. Rob's current interests lie in the development and implementation of self-tuning, real-time gas turbine models for diagnostic and virtual sensing applications.*

Don Simon *received a B.S. degree from Youngstown State University (1987) and an M.S. degree from Cleveland State University (1990) both in electrical engineering. He has been employed by the US Army Research Laboratory (1987 - present). During this time he has worked in the Controls and Dynamics Technology Branch of the NASA Glenn Research Center which is involved in the development of advanced control and health management technologies for current and future aerospace propulsion systems. Mr. Simon's specific areas of work include aircraft engine performance diagnostics, control design, and real-time control implementation.*